

# Cheatsheet avancé

Pour aller un peu plus loin quand on commence à dompter la bête...

## Gérer fichiers et répertoires

Créer un répertoire (**make directory**):  
`mkdir rép`

Créer des répertoires imbriqués:  
`mkdir -p rép1/rép2`

Changer de répertoire (**change dir**):  
`cd nouveau_rép`  
`cd ..` (répertoire parent)  
`cd` (répertoire personnel)  
`cd ~alice` (répertoire personnel de `alice`)

Afficher répertoire courant (**print working dir**):  
`pwd`

**Copier** un fichier vers un autre:  
`cp fichier_orig fichier_dest`

Copier des fichiers dans un répertoire:  
`cp fichier1 fichier2 rép`

Copier des répertoires entiers (**recursively**):  
`cp -r rép_orig rép_dest`  
`rsync -a rép_orig/ rép_dest/`

Créer un lien symbolique:  
`ln -s fichier_orig lien`

Renommer un fichier, lien ou répertoire:  
`mv fichier_orig fichier_dest`

Supprimer (**remove**) des fichiers ou des liens:  
`rm fichier1 fichier2`

Supprimer un répertoire (**remove dir**):  
`rmdir rép`

Supprimer un répertoire non vide (**force**):  
`rm -rf rép`

### Afficher les noms des fichiers

Énumérer (**list**) les fichiers «ordinaires» (ne commençant pas par `.`) dans le rép. courant:  
`ls`

Afficher une liste détaillée (**long**):  
`ls -l`

Énumérer tous (**all**) les fichiers dans le rép. courant (y compris ceux commençant par `.`):  
`ls -a`

Trier par date (**time**) (d'abord les plus récents):  
`ls -t`

Trier par taille (**size**) (d'abord les plus gros)  
`ls -S`

Affichier en inversant (**reverse**) l'ordre de tri:  
`ls -r`

### Afficher le contenu des fichiers

Afficher bout à bout le contenu de fichiers:  
`cat fichier1 fichier2` (concatenate)

Afficher le contenu de plusieurs fichiers (en

faisant une pause à chaque page):  
`more fichier1 fichier2`  
`less fichier1 fichier2` (plus de possibilités)

Afficher les 10 premières lignes d'un fichier:  
`head -10 fichier`

Afficher les 10 dernières lignes d'un fichier:  
`tail -10 fichier`

### Modèles de noms de fichiers

`*` tous les fichers  
`.*` tous les fichiers "cachés"  
`*.log` fichiers finissant par `'.log'`  
`*bug*` fichiers avec `'bug'` dans leur nom  
`*.*` fichiers finissant par `'.'` et suivis d'un caractère

### Gérer le contenu des fichiers

N'afficher que les lignes d'un fichier contenant une expression donnée:  
`grep expression fichier`

Recherche insensible aux majusc. / minusc.:  
`grep -i expression fichier`

Afficher toutes les lignes sauf celles qui contiennent une expression:  
`grep -v expression fichier`

Recherche récursive dans un rép.:  
`grep -r expression rép`

Trier les lignes d'un fichier:  
`sort fichier`

Trier, n'afficher qu'1 fois les lignes identiques:  
`sort -u fichier` (ou `uniq`)  
`uniq -c` (id. et compte chaque élément)

### Droits d'accès aux fichiers

Ajouter droits en écriture au propriétaire:  
`chmod u+w fichier` (user, write)

Ajouter droits en lecture au groupe du fichier:  
`chmod g+r fichier` (group, read)

Ajouter droits d'exécution aux autres utilisat.:  
`chmod o+x fichier`

Ajouter droits lecture / écriture à tous (**all**):  
`chmod a+rw fichier`

Rendre fich. exécutables exécutables par tous:  
`chmod a+rx *`

Rendre le répertoire et tous les fichiers qu'il contient accessibles par tous les utilisateurs:  
`chmod -R a+rx rép` (recursive)

Changer le propriétaire et le groupe d'un répertoire et tout ce qu'il contient:  
`chown -R nouvproprio:nouvgroupe rép`

### Comparer: fichiers, répertoires

Comparer 2 fichiers:  
`diff fichier1 fichier2`

Comparer 2 répertoires:  
`diff -r rép1 rép2`

### Rechercher des fichiers

Rechercher tous les fichiers dans le répertoire courant (`.`) avec `log` dans leur nom:  
`find . -name "*log"`

Trouver tous les fichiers en `.pdf` dans `rép` et exécuter une commande sur chacun:  
`find . -name "*.pdf" -exec xpdf {} ';' ,`

Recherche rapide dans tout système: (utilise un index, les fichiers récents peuvent manquer):  
`locate "bar"`

### Rediriger sortie de commande

Rediriger sortie de commande vers un fichier:  
`ls *.png > fichiers_image`

Ajouter la sortie d'une commande à un fichier:  
`ls *.jpg >> fichiers_image`

Rediriger la sortie d'une commande vers l'entrée d'une autre:  
`cat *.log | grep erreur`

### Contrôle de tâches

Afficher tous les processus exécutés:  
`ps -ef`

Classement en direct des processus (**P, M, T**):  
trie par utilisation Proc, Mém ou Temps CPU):  
`top`

Afficher la mémoire libre:  
`free`

Afficher la mémoire virtuelle toutes les n secs:  
`vmstat n`

Envoyer un signal d'arrêt à un processus:  
`kill <pid>` (numéro indiqué par `ps`)

Faire tuer un processus par le système:  
`kill -9 <pid>`

Tuer tous processus que l'on a le droit de tuer:  
`kill -9 -1`

Tuer tous processus appelés `windows`:  
`killall windows`

Tuer une application en mode graphique:  
`xkill` (cliquer sur la fenêtre du programme)

### Taille de fichiers et partitions

Afficher l'espace total occupé sur le disque par des fichiers ou des répertoires (**disk usage**):  
`du -sh rép1 rép2 fichier1 fichier2`

Nombre de caractères, mots et lignes:  
`wc fichier` (word count)

Afficher la taille, le **Type**, l'espace total et l'espace libre dans la partition courante:  
`df -Th .`

Afficher cette info pour toutes les partitions:  
`df -Th`

### Compresser

Compresser un fichier:  
`bzip2 fichier` (meilleur taux de compression)  
`gzip fichier`

Décompresser un fichier:  
`bunzip2 fichier.bz2`  
`gunzip fichier.gz`

### Manipuler des archives

Créer une archive compressée (**tape archive**)  
`tar jcvf archive.tar.bz2 rép/` (le mieux!)  
`tar zcvf archive.tar.gz rép/`

Tester (lister) une archive compressée:  
`tar jtvf archive.tar.bz2`  
`tar ztvf archive.tar.gz`

Extraire les fichiers d'une archive compressée:  
`tar jxvf archive.tar.bz2`  
`tar zxvf archive.tar.gz`

Options de **tar**:  
**c**: créer  
**t**: tester / lister  
**x**: extraire  
**j**: (dé)compression `bzip2` à la volée  
**z**: (dé)compression `gzip` à la volée

Manipuler des archives zip:  
`zip -r archive.zip <files>` (créer)  
`unzip -t archive.zip` (tester / lister)  
`unzip archive.zip` (extraire)

### Imprimer

Envoyer fichiers PostScript ou texte sur **queue**:  
`lpr -Pqueue f1.ps f2.txt` (local **p**rinter)

Lister les tâches d'impression dans **queue**:  
`lpq -Pqueue`

Annuler la tâche n°123 dans **queue**:  
`cancel 123 queue`  
`lprm -Pqueue 123`

### Gestion des utilisateurs

Afficher les utilisateurs connectés au système:  
`who`

Afficher sous quel utilisateur je suis connecté:  
`whoami`

Afficher à quel groupe appartient **utilisateur**:  
`groups utilisateur`

Passer à l'utilisateur **hulk**:  
`su - hulk`

Passer au super-utilisateur (**root**):  
`su -`  
`sudo -i`  
`su` (sans changer de rép. ni d'environnement)

Exécuter une commande en tant que **root**:  
`sudo commande`

### Gérer le temps

Attendre 60 secondes:  
`sleep 60`

Afficher la date actuelle:  
`date`

Mettre à l'heure ntp:  
`ntpdate time.nist.gov`

Mesurer le temps pris par une commande:  
`time trouve_prince_charmant -beau -riche`

### Aide sur les commandes

Aide de base (pour la plupart des commandes):  
`commande --help` (parfois `-h`)

Voir le manuel complet d'une commande:  
`man grep`  
`help cd` (commandes internes du shell)

### Réseau

Vérifier l'état d'un lien Ethernet:  
`ethtools <interface>`

Afficher toutes les interface réseau disponibles:  
`ip link show`

Assigner une adresse IP à une interface réseau:  
`ip address add 207.46.130.108 dev eth0`

Désactiver une interface réseau:  
`ip link set dev eth0 down`

Définir une passerelle par défaut:

`ip route add default via 192.168.0.1`

Supprimer la route par défaut:  
`ip route del default`

Tester la connexion réseau avec une machine:  
`ping 207.46.130.108`

### Bases d'administration système

Redémarrer la machine dans 5 minutes:  
`shutdown -r +5`

Eteindre la machine immédiatement:  
`shutdown -h now` (ou `halt`)

### Gestion des filesystems

Créer ou supprimer des partitions sur le premier disque IDE:  
`fdisk /dev/hda1`

Créer (formater) un système de fichiers ext3:  
`mkfs.ext3 /dev/hda1`

Créer (formater) un système de fichiers FAT32:  
`mkfs.vfat -v -F 32 /dev/hda2`

Monter une partition formatée:  
`mkdir /mnt/cleusb` (nécessaire une seule fois)  
`mount /dev/ubal /mnt/cleusb`

Monter image de système de fichiers (loopback)  
`mount -o loop initrd.img /mnt/initrd`

Utiliser un fichier comme un block device  
`losetup /dev/loop0 fichier`  
`losetup -d /dev/loop0` (libérer le device)

Démonter un système de fichiers:  
`umount /mnt/cleusb`

Créer un volume RAID:  
`mdadm -C /dev/md0 -l1 -n2 /dev/hda /dev/hdb`

Visualiser l'état d'un volume RAID:  
`mdadm -D /dev/md0`

Marquer un disque comme défaillant:  
`mdadm /dev/md0 -f /dev/hda`

Sortir un disque du volume RAID:  
`mdadm /dev/md0 -r /dev/hda`

Ajouter un disque dans un volume:  
`mdadm /dev/md0 -a /dev/hda`

Préparer un volume physique pour LVM:  
`pvccreate blockdev1 ... blockdevN`

Créer un volume group LVM:  
`vgcreate monvg blockdev1 ... blockdevN`

Créer un volume logique LVM de 30Mb:  
`lvcreate -n voll1 -L30M monvg`

Etendre un volume logique de 5Mb:  
`lvextend -L5M /dev/mapper/monvg/voll1`

Réduire un volume logique de 15Mb:  
`lvreduce -L15M /dev/mapper/monvg/voll1`

Créer un volume TrueCrypt (fichier ou device):  
`truecrypt -c`

(Dé)Monter un volume TrueCrypt:  
`truecrypt volume /mnt/crypto (-d)`

Créer un volume luks:  
`cryptsetup luksFormat /dev/ice`

Ouvrir un volume luks (monter ensuite `cryptdv`):  
`cryptsetup luksOpen /dev/ice cryptdv`

## Expressions régulières

dans *grep*, *vi*, *sed*, ... les caractères `?`, `+`, `{`, `|`, `(`, `)` doivent être précédés d'un antislash (`\`)

```
.      n'importe quel caractère
*      caractère précédent 0 fois ou +
+      caractère précédent 1 fois au moins
?      le caractère précédent 0 ou 1 fois
{n}    le caractère précédent n fois
{m,n}  le caractère précédent n fois ou plus
{n,}   le caractère précédent n fois ou plus
[a-z]   un caractère en minuscule
[a-zA-Z] une lettre
[0-9]   un chiffre
^/$     le début/la fin de ligne
|       séparateur (ou logique)

[[[:alnum:]]]  [[[:alpha:]]] ou [[[:digit:]]]
[[[:digit:]]]  [0-9]
[[[:alpha:]]]  [[[:upper:]]] ou [[[:lower:]]]
[[[:space:]]]  espace ou tabulation
[[[:lower:]]]  [a-z] ([[[:upper:]]] : [A-Z])
[[[:print:]]]  caractère imprimable ou espace
[[[:xdigit:]]] [A-Fa-f0-9]
```

## Raccourcis bash

```
<ctrl-a>  début de ligne
<ctrl-e>  fin (end) de ligne
<ctrl-→>  mot suivant
<ctrl-←>  mot précédent
<ctrl-k>  couper (kill) la fin de la ligne
<ctrl-y>  coller (yank)
<ctrl-d>  efface (delete) le caractère à droite du curseur
<esc>d    efface le mot à droite du curseur
<ctrl-t>  transposer deux caractères
<esc>t    transpose le mot précédent et le suivant
<ctrl-r>  rechercher dans l'historique
<ctrl-l>  effacer l'écran (équivalent à clear)
<ctrl-d>  envoie le caractère 'fin de fichier'
<ctrl-c>  arrêter la commande en cours
<ctrl-z>  stopper la commande en cours
<tab>     complète l'élément
<tab><tab> affiche les complétions possibles
```

## Historique et job control bash

```
!!      ré-exécuter la dernière commande
!txete  exécuter la dernière commande commençant par texte
!n     exécuter la commande numéro n
cmd &   exécuter la commande cmd en arrière plan
fg %n  mettre le travail n en avant plan (sans argument : le dernier travail)
bg %n  mettre le travail n en arrière plan (sans argument : le dernier travail)
kill %n terminer le travail n
jobs    affiche la liste des travaux du shell
history affiche l'historique des commandes
```

## Raccourcis less (ou man)

```
<espace>  avance d'une page
<return>  avance d'une ligne
<ctrl-return> avance de n lignes
b         recule (back) d'une page
nb        recule de n lignes
q         quitter less
/texte    rechercher la chaine texte
n         rechercher l'occurrence suivante
>         va au début du fichier
<         va à la fin du fichier
e         edite le fichier en cours
v         exécute command dans un shell
!command
```

# Vi

## Modes

```
i (<ins>) mode insertion
<esc>    mode commande
o        ajouter une ligne après la ligne courante et passer en insersion
```

## Sortir, sauver

```
:w        sauver le fichier
:w nom    sauver dans le fichier nom
:w>>nom   sauver à la fin du fichier nom
:q        quitter sans sauver
:q!       quitter de force sans sauver
:x        sauver et quitter
:e nom    ferme le fichier courant et édite le fichier nom
:e! nom   ferme le fichier courant de force et édite le fichier nom
:e!        annule tous les changements et revient au fichier original
```

## Déplacements

```
h        déplacement vers la droite
j        déplacement vers le bas
k        déplacement vers le haut
l        déplacement vers la gauche
^        début de ligne
$        fin de ligne
gg        première ligne (ou 1G)
G        dernière ligne
nG        ligne n (ou :n)
w        mot suivant
b        mot précédent
<ctrl-f> page suivante
<ctrl-b> page précédente
fx        aller après le prochain caractère x de la ligne
tx        aller avant le prochain caractère x de la ligne
H        aller en haut de l'écran
M        aller au milieu de l'écran
L        aller en bas de l'écran
```

## Copier, couper, coller

```
x        coupe le caractère sous le curseur
yy        copie la ligne courante entière
y^        copie la ligne du début au curseur
y$        copie la ligne du curseur à la fin
yw        copie le mot suivant
dd        coupe la ligne courante entière
d^        coupe la ligne du début au curseur
d$        coupe la ligne du curseur à la fin
dw        coupe le mot suivant
P         colle avant la position courante
p         colle après la position courante
```

## Numérations

*vi permet d'appliquer une commande à *n* zones*

```
nny      copie les n lignes suivantes
ndd      coupe les n lignes suivantes
nyw      coupe les n mots suivants
ndw      coupe les n mots suivants
...
```

## Rechercher

```
/motif    recherche motif vers le bas
?motif    recherche motif vers le haut
n         recherche la prochaine occurrence
p         recherche l'occurrence précédente
```

```
*         aller à la prochaine occurrence du mot sous le curseur
#         aller à l'occurence précédente du mot sous le curseur
```

## Remplacer

```
:m,ns/a/b/g remplace a par b entre les lignes m et n
:l,$s/a/b/g remplace a par b dans tout le fichier
:s/a/b/g    idem
```

## Signets

```
mx        crée le signet x
'x        va au signet x
d'x        coupe jusqu'au signet x
'.'        aller à la dernière ligne éditée
```

## Macros

```
qx        enregistre la macro x
q         termine l'enregistrement de la macro
@x        invoque la macro x
n@x       invoque n fois la macro x
```

## Registres

```
:reg       liste les registres et leur contenus
"*n        sélectionne le registre n
"*nd$      coupe la fin de la ligne courante et la sauve dans le registre n
"*np       colle le contenu du registre n
```

## Shell

```
:!cmd     exécuter cmd dans un shell
:!!cmd     remplace la totalité du buffer par le résultat de la commande cmd exécutée sur le buffer
:!rcmd    insère le résultat de cmd
!'xcmd    exécute la commande shell cmd sur le texte jusqu'au signet x démarre un shell sous vi
```

## Fenêtres (splits)

*les commandes :q, :w, ... s'appliquent dans chaque fenêtre*

```
:sp fic    coupe (split) l'écran en deux et charge fic dans la 2° moitié
:vsp fic   coupe l'écran en deux verticalement et charge fic dans la 2° moitié
```

*les commandes ci-dessous doivent être précédées de <ctrl-w>*

```
<ctrl-w> passe au split suivant
k         passe au split supérieur
j         passe au split inférieur
+         augmente la taille du split d'1 ligne (n<ctrl-w>+ augmente la taille du split de n lignes)
-         diminue la taille du split d'1 ligne
_         maximise le split courant
s         coupe l'écran en deux avec le contenu de la fenêtre courante
ajoute un split vide
```

## Divers

```
u         annuler
<ctrl-r>  refaire
<ctrl-l>  redessiner l'écran
<ctrl-g>  affiche des informations sur le fichier
<ctrl-n>  propose une complétion sur un mot
```

```
(en mode insertion)
<ctrn-x><ctrl-l> propose une complétion pour la ligne entière (en mode insertion)
<ctrl-r>=op insère le résultat de l'opération (mode insertion)
J         fusionner deux lignes
K         voir la page de man du mot sous le curseur
ga        affiche la valeur ascii du caractère
gf        ouvre le fichier sous le curseur
<ctrl-a>  incrémente le nombre sous le curseur
<ctrl-x>  décrémente le nombre sous le curseur
:help     aide
c         comme d, mais passe en mode insertion
.         répète la dernière commande
:Ex       ouvre l'explorateur de fichiers
:Sex      ouvre l'explorateur de fichiers dans un split
```

## Invoquer vi

Apprendre vi  
`vimtutor fr`

Ouvrir un fichier distant  
`vi scp://user@host//path/to/file`  
`vi ftp://user@host/path/to/file`

Multi-file edit (passer d'un fichier au suivant par `:n`)  
`vi fichier1 fichier2 ...`

Utiliser la version graphique de vim  
`gvim`

## Variables .vimrc

Utiliser l'indentation automatique  
`set autoindent`

Taille de l'indentation  
`set tabstop=4`  
`set shiftwidth=4`

Affichage de la parenthèse/ du crochet/de l'accolade correspondant  
`set showmatch`

Utilise la cloche visuelle plutôt que le beep  
`set noerrorbells`  
`set vb`

Affiche/n'affiche pas les correspondances de recherches en surbrillance  
`set hls`  
`set nohls`

Recherche incrémentale en cours de frappe  
`set incsearch`

Permet à vim de fonctionnel en uode non-compatible vi  
`set nocompatible`

Utilise la coloration syntaxique  
`syntax on`

Définit les raccourcis <ctrl-j> et <ctrl-k> pour passer d'une fenêtre à l'autre en la maximisant  
`map <C-J> <C-W>j<C-W>_`  
`map <C-K> <C-W>k<C-W>_`

Définit le raccourci <F10> pour enregistrer le fichier et quitter  
`map <F10> :wq<CR>`

# Emacs

## Sortir, sauver

*les commandes de cette section sont précédées de <ctrl-x>*

```
<ctrl-f>  ouvrir un fichier
<ctrl-s>  sauver le fichier
<ctrl-w>fi enregistrer sous fi
<ctrl-c>  quitter
```

## Déplacements

```
<ctrl-a>  début de ligne
<ctrl-e>  fin (end) de ligne
<ctrl-→>  mot suivant
<ctrl-←>  mot précédent
<esc> <   début du fichier
<esc> >   fin du fichier
```

## Copier, couper, coller

```
<ctrl-espace> pose une marque de début
<ctrl-k>       couper (kill) la fin de la ligne
<ctrl-y>       coller (yank)
<ctrl-w>       coupe de la marque au curseur
<ctrl-d>       efface (delete) le caractère à droite du curseur
<esc>d         efface le mot à droite du curseur
```

## Numération

`<ctrl-u>n cmd` exécute *n* fois *cmd*

## Rechercher

```
<ctrl-s>motif rechercher (search) motif
<ctrl-s>      recherche l'occurrence suivante
<ctrl-r>      motif recherche arrière
```

## Remplacer

```
<esc>%      remplacer
en réponse à la question : y remplace, n ne remplace pas, l remplace toutes les occurrences
```

## Shell

```
<esc>xshell démarre un shell
<esc>!cmd   exécute cmd
<esc>|cmd   exécute cmd sur le contenu de la région marquée
```

## Buffers

```
<ctrl-x>b   passe au buffer suivant
<ctrl-x>bnom passer au buffer nom
<ctrl-x>k   supprimer le buffer courant
```

## Fenêtres

```
<ctrl-x>2   diviser l'écran en 2 fenêtres
<ctrl-x>o   aller dans l'autre (other) fenêtre
<ctrl-x>1   1 seule fenêtre dans l'écran
```

## Divers

```
<ctrl-x>u   annuler la dernière commande
<ctrl-g>    annuler une commande en cours
<ctrl-t>    transposer deux caractères
<esc>t       transpose le mot précédent et le suivant
<ctrl-l>    centre l'écran sur la ligne courante
```